

Procedimiento selectivo convocado por Resolución de 20 de febrero de 2025 de la Dirección General de Recursos Humanos

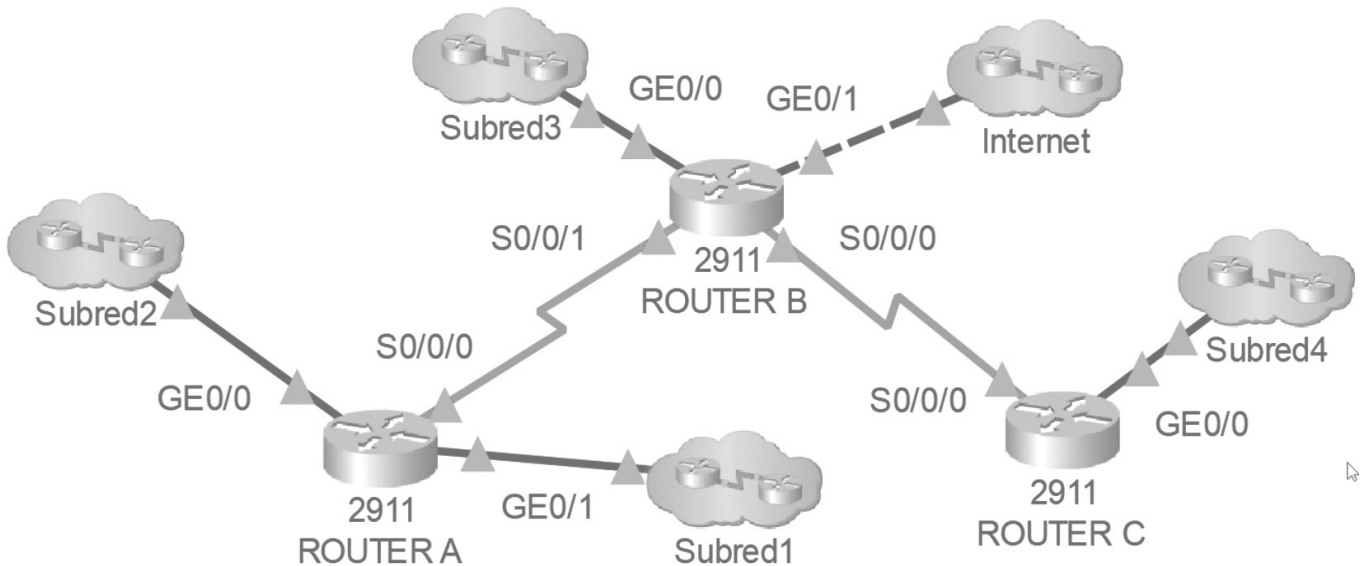
Cuerpo de Profesores de Enseñanza Secundaria especialidad Informática 590107

**DESARROLLO DE LA PRIMERA PRUEBA – PARTE A DE LA OPOSICIÓN
PRUEBA PRÁCTICA**

30 de junio de 2025

PRIMER EJERCICIO – 3.5 puntos

La siguiente imagen muestra la red de una empresa. Se ha obtenido con Packet Tracer. El número “2911” es el modelo de los enrutadores. No es relevante para la solución del ejercicio.



Se pide:

- a) **(1.5 puntos)** A partir de la red 192.168.128.0 con máscara 255.255.252.0, realizar el diseño de la red de tal forma que el tamaño de cada subred sea lo más ajustado posible. Debe tenerse en cuenta que las subredes cuentan con:
- Subred 1: Actualmente tiene 12 PC y debe reservarse espacio para un crecimiento del 25%.
 - Subred 2: Actualmente tiene 30 PC.
 - Subred 3: Actualmente tiene 5 PC y 4 servidores.
 - Subred 4: Actualmente tiene 300 PC y debe reservarse espacio para un crecimiento del 10%.

Para cada subred especifique la dirección de red, la máscara de subred y la dirección de difusión.

- b) **(0.5 puntos)** Introduzca los comandos necesarios para configurar el ‘Router A’ utilizando el protocolo RIP versión 2 para que genere dinámicamente su tabla de enrutamiento.
- c) **(1.5 puntos)** Indique para cada uno de los tres enrutadores cuál sería su tabla de enrutamiento una vez alcanzado en todos ellos el estado de convergencia tras haber sido todos configurados con el protocolo RIP versión 2.

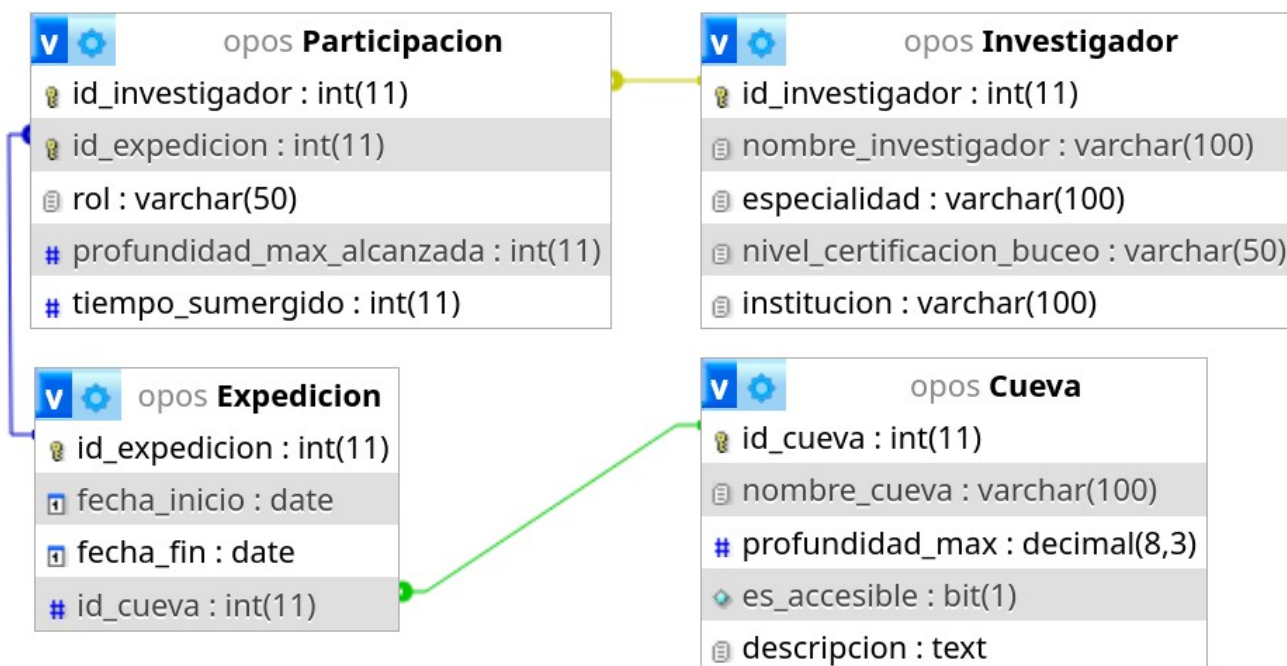
| Destino | Máscara | Interfaz | Métrica |
|---------|---------|----------|---------|
|---------|---------|----------|---------|

SEGUNDO EJERCICIO – 3 puntos

Para resolver este ejercicio solo se podrá utilizar la sintaxis de SQL ANSI Standard, Oracle o MySQL (a partir de la versión 8), no pudiendo mezclarse entre ellas en la resolución de todo el ejercicio. Se debe indicar al inicio de la solución qué sintaxis se usa.

Se deberá dar una única solución para cada apartado. De aparecer varias soluciones en un mismo apartado, solo se considerará la primera.

Se parte del siguiente modelo de datos (diagrama obtenido con phpMyAdmin):



Y se tiene la siguiente semántica que puede no aparecer reflejada en el modelo:

- La descripción de una cueva es un texto largo cuyo tamaño excede del límite de los char, varchar y varchar2.
- El campo es_accesible de una cueva almacena 1 o 0, donde 1 significa cierto y 0 falso. Por defecto, todas las cuevas son accesibles. Se podrá utilizar otro tipo de dato apropiado.
- La profundidad máxima de una cueva se expresa en metros. Es un número positivo o 0.
- Los nombres del investigador, de la cueva, y la fecha de inicio de la expedición son datos obligatorios.

- El nombre de una cueva y el nombre de un investigador no se pueden repetir en toda la base de datos.
- La fecha de fin de una expedición, en caso de conocerse, debe ser posterior a la fecha de inicio.
- Toda expedición se realiza sobre una cueva.
- No se podrán borrar las expediciones de las que se hayan guardado los investigadores que participan. Pero si se pueden eliminar investigadores que hayan participado en expediciones, borrando por tanto su participación.
- No se podrán borrar las cuevas para las que exista alguna expedición.
- El tiempo que permanece un investigador sumergido en una expedición se contabiliza en segundos, por lo que debe ser superior a 0.
- La profundidad máxima alcanzada por un investigador en una expedición, que se expresa en metros, debe ser igual o superior a 0, y no puede ser mayor que la profundidad máxima de la cueva.

Se pide:

- a) **(1.5 puntos)** Escribir el script de creación para la base de datos del modelo, con la limitación de no poder utilizar sentencias de tipo ALTER. En Oracle, debes considerar los tipos de datos equivalentes.
- b) Realizar las siguientes consultas sobre la base de datos:
 1. **(0.5 puntos)** Devuelve los nombres de los investigadores cuyo nombre contiene la letra "a" (mayúscula o minúscula) y que, en alguna expedición, han alcanzado una profundidad que esté dentro de los últimos 5 metros respecto a la profundidad máxima de la cueva explorada.
 2. **(0.5 puntos)** Nombre de las cuevas exploradas por al menos un investigador de cada especialidad existente.
 3. **(0.5 puntos)** Parejas de investigadores que han coincidido en al menos 2 expediciones distintas. Se mostrará el nombre de cada investigador, y el número de expediciones que ha realizado la pareja. Aparecerá primero la pareja que haya coincidido en más expediciones.

TERCER EJERCICIO – 3,5 puntos

Se pide escribir un programa que, a partir de un fichero de configuración en texto plano, genere una serie de matrices que se almacenarán también en ficheros de texto plano. El programa se puede escribir en Java, Python o PHP, sin mezclar lenguajes.

Formato del fichero de entrada

El fichero estará compuesto por una o más líneas, cada una de ellas con los datos necesarios para crear una matriz. Hay cinco opciones disponibles:

- <FILAS> <COLUMNAS>: crea una matriz con el número especificado de filas y columnas. Cada elemento de la matriz se inicializará aleatoriamente entre 0 y 1.
- <DIM> SIM: crea una matriz simétrica con DIM filas y DIM columnas. Los de la matriz se inicializarán aleatoriamente entre 0 y 1, pero de manera que el elemento (i, j) sea igual al (j, i) para todos los casos.
- <FILAS> <COLUMNAS> <MIN> <MAX>: crea una matriz con el número especificado de filas y columnas, cada elemento de la matriz se inicializará aleatoriamente entre MIN y MAX.
- <DIM> <MIN> <MAX> SIM: crea una matriz simétrica de DIM filas y DIM columnas. Los elementos de la matriz se inicializarán aleatoriamente entre MIN y MAX, pero de manera que el elemento (i, j) sea igual al (j, i) para todos los casos.
- <FILAS> <COLUMNAS> <NUM>: crea una matriz con el número especificado de filas y columnas en la que todos los elementos tienen el valor NUM.

<FILAS>, <COLUMNAS> y <DIM> tienen que ser números enteros positivos. <MIN> y <MAX> son números reales y se debe cumplir que <MIN> sea menor que <MAX> (en una misma línea). <NUM> también es un número real.

Por ejemplo, el siguiente fichero tiene un formato correcto:

```
3 SIM
2 3
1 4 2.5 3.5
2 6.05 7.05 SIM
2 3 5.0
```

Si alguna de las líneas del fichero no se ajusta a las opciones disponibles, el fichero se considerará no válido.

Parámetros de entrada

El programa recibirá tres argumentos:

- El fichero de entrada: En Java y Python, el programa recibirá el nombre del fichero como primer argumento. En PHP, recibirá el fichero en `$FILES['confi']`.
- Una cadena con el nombre base para los ficheros de salida. En Java y Python, el programa recibirá el nombre base como segundo argumento. En PHP, lo recibirá en `$POST['base']`.
- Una cadena con la extensión de los ficheros de salida. En Java y Python, el programa recibirá la extensión como tercer argumento. En PHP, la recibirá en `$POST['exten']`.

Validación del fichero de entrada (0.75 puntos)

Un fichero es válido si todas sus líneas se ajustan a una de las opciones enumeradas anteriormente.

El programa debe comprobar que el formato del fichero es válido usando una función o método llamado `hayErrores()` que devolverá 0 si no hay errores en el fichero o el número de línea en la que se encuentra el primer error. Recibirá como argumento el contenido del fichero.

Requisitos de salida (2 puntos)

Si el fichero es válido, por cada línea del mismo se creará un nuevo fichero que almacenará una matriz en formato texto. Cada uno de estos ficheros tendrá como nombre el nombre base recibido como parámetro, concatenado con el número de línea y con la extensión recibida como parámetro.

Las matrices se almacenarán en texto plano, con una fila por línea y los elementos separados por espacios. Los elementos de las matrices serán siempre números reales.

Por ejemplo, partiendo del siguiente fichero:

```
3 SIM
2 3
1 4 2.5 3.5
2 6.05 7.05 SIM
2 3 5.0
```

con nombre base “matriz” y extensión “txt” se generarían los siguientes ficheros (hay que tener en cuenta que excepto en el último caso se trata de números aleatorios de ejemplo):

| matriz1.txt | matriz2.txt | matriz3.txt | matriz4.txt | matriz5.txt |
|-------------|-------------|-----------------|-------------|-------------|
| 0.1 0.2 0.4 | 0.1 0.4 0.5 | 2.5 3.4 3.1 2.8 | 6.1 6.8 | 5.0 5.0 5.0 |
| 0.2 0.3 0.7 | 0.9 0.8 0.6 | | 6.8 6.9 | 5.0 5.0 5.0 |
| 0.4 0.7 0.5 | | | | |

Además de crear los ficheros, el programa mostrará un mensaje de confirmación indicando el número de ficheros creados.

Si el fichero de entrada no es válido, no se generará ningún fichero de salida.

Control de errores (0.75 puntos)

Se mostrarán mensajes de error apropiados si:

- El fichero de entrada no es válido (el mensaje de error incluirá el número de línea del primer error).
- Hay algún error en los parámetros de entrada.
- El fichero de entrada no se puede leer o está vacío.
- No se pueden crear los ficheros de salida.